

[0001] A FAST JOINT DETECTION USER EQUIPMENT

[0002] This application is a continuation of U.S. Patent Application No. 10/037,710, filed December 31, 2001, which claims priority to U.S. Provisional Patent Application No. 60/287,431, filed on April 30, 2001, which applications are incorporated herein by reference.

[0003] BACKGROUND

[0004] The invention generally relates to wireless communication systems. In particular, the invention relates to data detection in a wireless communication system.

[0005] Figure 1 is an illustration of a wireless communication system 10. The communication system 10 has base stations 12₁ to 12₅ (12) which communicate with user equipments (UEs) 14₁ to 14₃ (14). Each base station 12 has an associated operational area, where it communicates with UEs 14 in its operational area.

[0006] In some communication systems, such as frequency division duplex using code division multiple access (FDD/CDMA) and time division duplex using code division multiple access (TDD/CDMA), multiple communications are sent over the same frequency spectrum. These communications are differentiated by their channelization codes. To more efficiently use the frequency spectrum, TDD/CDMA communication systems use repeating frames divided into timeslots for communication. A communication sent in such a system will have one or multiple associated codes and timeslots assigned to it.

[0007] Since multiple communications may be sent in the same frequency spectrum and at the same time, a receiver in such a system must distinguish between the multiple communications. One approach to detecting such signals is multiuser detection (MUD). In MUD, signals associated with all the UEs 14, users, are detected simultaneously. Another approach to detecting a multi-code transmission from a single transmitter is single user detection (SUD). In SUD, to recover data from the multi-code transmission at the receiver,

the received signal is passed through an equalization stage and despread using one or multiple codes. Approaches for implementing MUD and the equalization stage of SUD include using a Cholesky or an approximate Cholesky decomposition. These approaches have a high complexity. The high complexity leads to increased power consumption, which at the UE 14 results in reduced battery life. Accordingly, it is desirable to have alternate approaches to detecting received data.

[0008]

SUMMARY

[0009] A code division multiple access user equipment receives K data signals over a shared spectrum. The user equipment receives and samples a combined signal having the K transmitted data signals. A combined channel response matrix is produced. A block column of a cross correlation matrix is determined using the combined channel response matrix. Each block entry of the block column is a K by K matrix. Each block entry of the block column is a K by K matrix. A fourier transform of a complex conjugate transpose of the combined channel response matrix multiplied to the combined signal samples is taken. An inverse of a fourier transform of each block entry is multiplied to a result of the fourier transform to produce a fourier transform of the data vector. An inverse fourier transform of the data vector fourier transform is taken to produce data of the K signals.

[0010]

BRIEF DESCRIPTION OF THE DRAWING(S)

- [0011] Figure 1 is a wireless communication system.
- [0012] Figure 2 is a simplified transmitter and a fast joint detection receiver.
- [0013] Figure 3 is an illustration of a communication burst.
- [0014] Figure 4 is a flow chart of a preferred embodiment for fast joint detection.
- [0015] Figure 5 is an illustration of a data burst indicating extended processing areas.
- [0016] Figures 6-11 are graphs illustrating the simulated performance of fast joint detection to other data detection approaches.

[0017] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0018] Figure 2 illustrates a simplified transmitter 26 and receiver 28 using fast joint detection in a TDD/CDMA communication system, although fast joint detection is applicable to other systems, such as FDD/CDMA. In a typical system, a transmitter 26 is in each UE 14 and multiple transmitting circuits 26 sending multiple communications are in each base station 12. The joint detection receiver 28 may be at a base station 12, UEs 14 or both.

[0019] The transmitter 26 sends data over a wireless radio channel 30. A data generator 32 in the transmitter 26 generates data to be communicated to the receiver 28. A modulation/spreading/training sequence insertion device 34 spreads the data with the appropriate code(s) and makes the spread reference data time-multiplexed with a midamble training sequence in the appropriate assigned time slot, producing a communication burst or bursts.

[0020] A typical communication burst 16 has a midamble 20, a guard period 18 and two data fields 22, 24, as shown in Figure 3. The midamble 20 separates the two data fields 22, 24 and the guard period 18 separates the communication bursts to allow for the difference in arrival times of bursts transmitted from different transmitters 26. The two data fields 22, 24 contain the communication burst's data.

[0021] The communication burst(s) are modulated by a modulator 36 to radio frequency (RF). An antenna 38 radiates the RF signal through the wireless radio channel 30 to an antenna 40 of the receiver 28. The type of modulation used for the transmitted communication can be any of those known to those skilled in the art, such as quadrature phase shift keying (QPSK) or M-ary quadrature amplitude modulation (QAM).

[0022] The antenna 40 of the receiver 28 receives various radio frequency signals. The received signals are demodulated by a demodulator 42 to produce a baseband signal. The baseband signal is sampled by a sampling device 43, such as one or multiple analog to digital converters, at the chip rate or a multiple of the chip rate of the transmitted bursts. The

samples are processed, such as by a channel estimation device 44 and a fast joint detection device 46, in the time slot and with the appropriate codes assigned to the received bursts. The channel estimation device 44 uses the midamble training sequence component in the baseband samples to provide channel information, such as channel impulse responses. The channel impulse responses for all the transmitted signals can be viewed as a matrix, H . The channel information is used by the fast joint detection device 46 to estimate the transmitted data of the received communication bursts as soft symbols.

[0023] The fast joint detection device 46 uses the channel information provided by the channel estimation device 44 and the known spreading codes used by the transmitter 26 to estimate the data of the desired received communication burst(s).

[0024] Although fast joint detection is explained using the third generation partnership project (3GPP) universal terrestrial radio access (UTRA) TDD system as the underlying communication system, it is applicable to other systems. That system is a direct sequence wideband CDMA (W-CDMA) system, where the uplink and downlink transmissions are confined to mutually exclusive timeslots.

[0025] The receiver 28 receives a total of K bursts that arrive simultaneously. The K bursts are superimposed on top of each other in one observation interval. For the 3GPP UTRA TDD system, each data field of a time slot corresponds to one observation interval. A code used for a k^{th} burst is represented as $C^{(k)}$. The K bursts may originate from K different transmitters or for multi-code transmissions, less than K different transmitters.

[0026] Each data field of a communication burst has a predetermined number of transmitted symbols, N_s . Each symbol is transmitted using a predetermined number of chips, which is the spreading factor, SF. Accordingly, each data field has $N_s \times SF$ chips. After passing through the wireless radio channel, each symbol has an impulse response, such as of length W chips. A typical value for W is 57. As a result, each received field has a length of $SF \times N_s + W - 1$ chips or N_c chips.

[0027] Each k^{th} field of the K data fields in one observation interval can be modeled at the receiver by Equation 1.

$$\underline{r}^{(k)} = A^{(k)} \underline{d}^{(k)}, k = 1 \dots K \quad \text{Equation 1}$$

[0028] $\underline{r}^{(k)}$ is the received contribution of the k^{th} field. $A^{(k)}$ is the combined channel response for the k^{th} field. $A^{(k)}$ is a $N_C \times N_S$ matrix. Each j^{th} column in $A^{(k)}$ is a zero-padded version of the symbol response $S^{(k)}$ of the j^{th} element of $\underline{d}^{(k)}$. The symbol response $S^{(k)}$ is the convolution of the k^{th} field's estimate response, $\underline{h}^{(k)}$, and spreading code $C^{(k)}$ for the field. $\underline{d}^{(k)}$ is the unknown data symbols in the k^{th} data field. $\underline{h}^{(k)}$ is of length W chips and can be represented by Equation 2.

$$\underline{h}^{(k)} = \gamma^{(k)} \cdot \tilde{\underline{h}}^{(k)} \quad \text{Equation 2}$$

$\gamma^{(k)}$ reflects the transmitter gain and path loss. $\tilde{\underline{h}}^{(k)}$ is the channel impulse response.

[0029] For uplink communications, each $\tilde{\underline{h}}^{(k)}$ as well as each $\gamma^{(k)}$ are distinct. For the

downlink, all of the fields have the same $\tilde{\underline{h}}^{(k)}$ but each $\gamma^{(k)}$ is different. If transmit diversity

is used in the downlink, each $\gamma^{(k)}$ and $\tilde{\underline{h}}^{(k)}$ are distinct.

[0030] The overall received vector \underline{r} from all K fields sent over the wireless channel is per Equation 3.

$$\underline{r} = \sum_{k=1}^K \underline{r}^{(k)} + \underline{n} \quad \text{Equation 3}$$

\underline{n} is a zero-mean noise vector.

[0031] By combining the $\underline{A}^{(k)}$ for all data fields into a total channel response matrix A and all the unknown data for each burst $\underline{d}^{(k)}$ into a total data vector \underline{d} , Equation 1 becomes Equation 4.

$$\underline{r} = A \underline{d} + \underline{n} \quad \text{Equation 4}$$

[0032] Determining \underline{d} using a MMSE solution is per Equation 5.

$$\underline{d} = R^{-1} \left(A^H \underline{r} \right) \quad \text{Equation 5}$$

$(\cdot)^H$ represents the hermetian function (complex conjugate transpose). R for a MMSE solution is per Equation 6.

$$R = A^H A + \sigma^2 I \quad \text{Equation 6}$$

σ^2 is the noise variance, typically obtained from the channel estimation device 44, and I is the identity matrix.

[0033] Using fast fourier transforms (FFTs), although other fourier transforms may be used, this equation is solved, preferably, per Equation 7.

$$[F(\underline{d})]_k = [\Lambda^{(k)}]^{-1} [F(A^H \underline{r})]_k \quad \text{Equation 7}$$

$F(\cdot)$ indicates the FFT function. The $[\cdot]_k$ indicate that the equation is solved at each frequency point k . $\Lambda^{(k)}$ are block entries of size K by K of a block diagonal matrix Λ . The derivation of Λ is described subsequently. Instead of directly solving Equation 7, Equation 7 can be solved using forward and backward substitution.

[0034] Figure 4 is a flow chart for a preferred method of determining the data vector \underline{d} using fast joint detection. The combined channel response matrix A is determined using the estimated responses $\underline{h}^{(k)}$ and the spreading code $\underline{c}^{(k)}$ for each burst $c^{(k)}$, 48. Form the combined channel correlation matrix, $R = A^H A$, 49. At each frequency point, a K by K matrix $\Lambda^{(k)}$ is determined by taking the fourier transform of block entries of a block column of R (block FFT), 50. Preferably, a central column is used, at least w columns from the left or right of the R matrix.

[0035] $F[A^H \underline{r}]_k$ is determined using a FFT of a matrix multiplication, 51. The inverse of each $\Lambda^{(k)}$, $[\Lambda^{(k)}]^{-1}$, is determined. To determine $[F(\underline{d})]_k$, $[\Lambda^{(k)}]^{-1}$ and $[F(A^H \underline{r})]_k$ are multiplied at each frequency point. Alternately, $[F(\underline{d})]_k$ is determined using LU decomposition. $\Lambda^{(k)}$ is decomposed into a lower triangular matrix, L , and an upper triangular matrix, U , 52. Using forward, $L \underline{y} = [F(A^H \underline{r})]_k$, 53, and backward substitution,

$U[F(\underline{d})]_K = \underline{y}$, 54, $[F(\underline{d})]_K$ is determined. \underline{d} is determined by an inverse FFT of $F(\underline{d})$, 55.

[0036] The derivation of Equation 7 is as follows. A minimum mean square error solution of Equation 4 is determined per Equation 8. Although Equation 7 is a MMSE based solution, fast joint detection can be implemented using other approaches, such as a zero forcing approach.

$$R\underline{d} = (A^H A + \sigma^2 I) \underline{d} = A^H \underline{r} \quad \text{Equation 8}$$

If a zero forcing solution is used, the $\sigma^2 I$ term drops out of Equation 8, such as $R\underline{d} = (A^H A) \underline{d} = A^H \underline{r}$. The following is a derivation for the MMSE solution, although an analogous derivation can be used for a zero forcing solution. For illustrative purposes, a simplified example of R with $N_s = 10$ and $W = 2$ is per Equation 9. This example is readily extendable to any N_s and W .

$$R = \begin{bmatrix} R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots \\ R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 & 0 \\ 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 \\ 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 \\ 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 \\ 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 \\ 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H \\ \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 \end{bmatrix}$$

Equation 9

[0037] The matrix R is of size (KN_s) by (KN_s) in general. Each entry, R_i , in the R matrix is a K by K block. The sub-matrix, within the dotted lines of R, is block-circulant, i.e. a block-wise extension of a circulant matrix. The portion of R, which is not block-circulant, depends on the maximum multipath delay spread, W.

[0038] A block-circulant extension of the matrix R, R_C , in Equation 9 is per Equation 10.

$$R_c = \begin{bmatrix} R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 \\ R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 & 0 & R_2 \\ \dots & \dots \\ R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 & 0 \\ 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 \\ 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 & 0 \\ 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 & 0 \\ 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H & 0 \\ 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H & {R_2}^H \\ \dots & \dots \\ {R_2}^H & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 & {R_1}^H \\ {R_1}^H & {R_2}^H & 0 & 0 & 0 & 0 & 0 & R_2 & R_1 & R_0 \end{bmatrix}$$

Equation 10

[0039] A "Discrete Fourier Transform (DFT)-like" matrix D is determined such that

$R_c = D \Delta D^H$. One such matrix D is per Equation 11.

$$D = \begin{bmatrix} I & e^{\frac{j2\pi}{N_s}} I_K & e^{\frac{j4\pi}{N_s}} I_K & \cdots & \cdots & e^{\frac{j18\pi}{N_s}} I_K \\ I & e^{\frac{j4\pi}{N_s}} I_K & e^{\frac{j8\pi}{N_s}} I_K & \cdots & \cdots & e^{\frac{j36\pi}{N_s}} I_K \\ \vdots & \vdots & \vdots & & & \vdots \\ I & e^{\frac{j18\pi}{N_s}} I_K & e^{\frac{j36\pi}{N_s}} I_K & \cdots & \cdots & e^{\frac{j162\pi}{N_s}} I_K \\ I & e^{\frac{j20\pi}{N_s}} I_K & e^{\frac{j40\pi}{N_s}} I_K & \cdots & \cdots & e^{\frac{j180\pi}{N_s}} I_K \\ I & e^{\frac{jN_s\pi}{N_s}} I_K & e^{\frac{jN_s\pi}{N_s}} I_K & \cdots & \cdots & e^{\frac{jN_s\pi}{N_s}} I_K \end{bmatrix}$$

Equation 11

I_K is a K by K identity matrix.

[0040] The product $D^H D$ is per Equation 9.

$$D^H D = N_s I_{KN_s} \quad \text{Equation 12}$$

I_{KN_s} is a KN_s by KN_s identity matrix. The block circulant matrix R_c is multiplied by the D matrix, such as per Equation 13.

$$R_c D = \begin{bmatrix} (R_0 + R_1^H + R_2^H + \dots + R_1 + R_2) e^{\frac{j2\pi}{N_s}} + R_0^H e^{\frac{j4\pi}{N_s}} + R_2^H e^{\frac{j6\pi}{N_s}} + \dots + R_0 e^{\frac{j18\pi}{N_s}} + R_1^H e^{\frac{j36\pi}{N_s}} + R_2^H e^{\frac{j54\pi}{N_s}} + \\ (R_0 + R_1^H + R_2^H + \dots + R_1 + R_2) e^{\frac{j18\pi}{N_s}} + R_2^H e^{\frac{j20\pi}{N_s}} + R_2 e^{\frac{j162\pi}{N_s}} + R_1 e^{\frac{j180\pi}{N_s}} \\ (R_0 + R_1^H + R_2^H + \dots + R_1 + R_2) e^{\frac{j2\pi}{N_s}} + R_1^H e^{\frac{j4\pi}{N_s}} + R_0^H e^{\frac{j6\pi}{N_s}} + \dots + R_1 e^{\frac{j18\pi}{N_s}} + R_0^H e^{\frac{j36\pi}{N_s}} + R_1^H e^{\frac{j54\pi}{N_s}} + \\ (R_0 + R_1^H + R_2^H + \dots + R_1 + R_2) e^{\frac{j8\pi}{N_s}} + R_2^H e^{\frac{j20\pi}{N_s}} + R_2 e^{\frac{j72\pi}{N_s}} + R_2 e^{\frac{j180\pi}{N_s}} \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \\ (R_0 + R_1^H + R_2^H + \dots + R_1 + R_2) e^{\frac{j12\pi}{N_s}} + R_1^H e^{\frac{j14\pi}{N_s}} + R_0^H e^{\frac{j16\pi}{N_s}} + \dots + R_0 e^{\frac{j108\pi}{N_s}} + R_1^H e^{\frac{j126\pi}{N_s}} + R_0^H e^{\frac{j144\pi}{N_s}} + \\ (R_0 + R_1^H + R_2^H + \dots + R_1 + R_2) e^{\frac{j18\pi}{N_s}} + R_1^H e^{\frac{j20\pi}{N_s}} + R_1^H e^{\frac{j162\pi}{N_s}} + R_2^H e^{\frac{j180\pi}{N_s}} \\ \vdots \qquad \qquad \qquad \vdots \qquad \qquad \qquad \vdots \end{bmatrix}$$

Equation 13

Each entry of $R_c D$ is a K by K block. A block-diagonal matrix Λ is per Equation 14.

$$\Lambda \stackrel{\Delta}{=} \begin{bmatrix} \Lambda^{(1)} & & & & \\ & \Lambda^{(2)} & & & \\ & & \Lambda^{(3)} & & \\ & & & \ddots & \\ & & & & \Lambda^{(N_s)} \end{bmatrix} \quad \text{Equation 14}$$

Λ is of size (KN_s) by (KN_s) . Each entry $\Lambda^{(i)}$ of Λ is per Equation 15.

$$\Lambda^{(i)} = \begin{bmatrix} \lambda_{11}^{(i)} & \dots & \lambda_{1K}^{(i)} \\ \vdots & & \vdots \\ \lambda_{K1}^{(i)} & \dots & \lambda_{KK}^{(i)} \end{bmatrix}.$$

Equation 15

$\Lambda^{(i)}$ is a K by K block and has K^2 non-zero entries.

[0041] The D matrix is multiplied to the Λ matrix, such as per Equation 16.

$$D\Lambda = \begin{bmatrix} \Lambda^{(1)} & \Lambda^{(2)} e^{\frac{j2\pi}{N_s}} & \Lambda^{(3)} e^{\frac{j4\pi}{N_s}} & \dots & \Lambda^{(N_s-1)} e^{\frac{j16\pi}{N_s}} & \Lambda^{(N_s)} e^{\frac{j18\pi}{N_s}} \\ \Lambda^{(1)} & \Lambda^{(2)} e^{\frac{j4\pi}{N_s}} & \Lambda^{(3)} e^{\frac{j8\pi}{N_s}} & \dots & \Lambda^{(N_s-1)} e^{\frac{j32\pi}{N_s}} & \Lambda^{(N_s)} e^{\frac{j36\pi}{N_s}} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ \Lambda^{(1)} & \Lambda^{(2)} e^{\frac{j14\pi}{N_s}} & \Lambda^{(3)} e^{\frac{j28\pi}{N_s}} & \dots & \Lambda^{(N_s-1)} e^{\frac{j112\pi}{N_s}} & \Lambda^{(N_s)} e^{\frac{j126\pi}{N_s}} \\ \Lambda^{(1)} & \Lambda^{(2)} e^{\frac{j16\pi}{N_s}} & \Lambda^{(3)} e^{\frac{j32\pi}{N_s}} & \dots & \Lambda^{(N_s-1)} e^{\frac{j128\pi}{N_s}} & \Lambda^{(N_s)} e^{\frac{j144\pi}{N_s}} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \end{bmatrix}$$

Equation 16

Each entry of $D\Lambda$ as shown in Equation 16 is a K by K block.

[0042] The system of equations produced by equating each row of $R_c D$ with each row of $D\Lambda$ is consistent. Accordingly, the same set of equations result by equating any row of $R_c D$ with the same row of $D\Lambda$. To illustrate for Equation 13, the first row-block of $R_c D$ is per Equation 17.

$$[(R_0 + R_1^H + R_2^H + R_1 + R_2), (R_0 e^{\frac{j2\pi}{N_s}} + R_1^H e^{\frac{j4\pi}{N_s}} + R_2^H e^{\frac{j6\pi}{N_s}} + R_2 e^{\frac{j18\pi}{N_s}} + R_1 e^{\frac{j20\pi}{N_s}}), \dots; \\ (R_0 e^{\frac{j16\pi}{N_s}} + R_1^H e^{\frac{j32\pi}{N_s}} + R_2^H e^{\frac{j48\pi}{N_s}} + R_2 e^{\frac{j144\pi}{N_s}} + R_1 e^{\frac{j160\pi}{N_s}}), (R_0 e^{\frac{j18\pi}{N_s}} + R_1^H e^{\frac{j36\pi}{N_s}} + R_2^H e^{\frac{j54\pi}{N_s}} + R_2 e^{\frac{j162\pi}{N_s}} + R_1 e^{\frac{j180\pi}{N_s}})]$$

Equation 17

[0043] The first row-block of $D\Lambda$ is per Equation 18.

$$[\Lambda^{(1)}, \Lambda^{(2)} e^{\frac{j2\pi}{N_s}}, \Lambda^{(3)} e^{\frac{j4\pi}{N_s}}, \dots, \Lambda^{(N_s-1)} e^{\frac{j16\pi}{N_s}}, \Lambda^{(N_s)} e^{\frac{j18\pi}{N_s}}] \quad \text{Equation 18}$$

[0044] Equating the entries of these two rows, Equations 19 and 20 result.

$$\Lambda^{(1)} = (R_0 + R_1^H + R_2^H + R_1 + R_2) \quad \text{Equation 19}$$

$$\Lambda^{(2)} e^{\frac{j2\pi}{N_s}} = \left(R_0 e^{\frac{j2\pi}{N_s}} + R_1^H e^{\frac{j4\pi}{N_s}} + R_2^H e^{\frac{j6\pi}{N_s}} + R_2 e^{\frac{j18\pi}{N_s}} + R_1 e^{\frac{j20\pi}{N_s}} \right) = e^{\frac{j2\pi}{N_s}} \left(R_0 + R_1^H e^{\frac{j2\pi}{N_s}} + R_2^H e^{\frac{j4\pi}{N_s}} + R_2 e^{\frac{-j4\pi}{N_s}} + R_1 e^{\frac{-j2\pi}{N_s}} \right)$$

Equation 20

[0045] As a result, $\Lambda^{(2)}$ is per Equation 21.

$$\Lambda^{(2)} = (R_0 + R_1^H e^{\frac{j2\pi}{N_s}} + R_2^H e^{\frac{j4\pi}{N_s}} + R_2 e^{\frac{-j4\pi}{N_s}} + R_1 e^{\frac{-j2\pi}{N_s}}) \quad \text{Equation 21}$$

[0046] Similarly, $\Lambda^{(N_s-1)}$ is per Equation 22.

$$\Lambda^{(N_s-1)} = (R_0 + R_1^H e^{\frac{j2(N_s-2)\pi}{N_s}} + R_2^H e^{\frac{j4(N_s-2)\pi}{N_s}} + R_2 e^{\frac{-j4(N_s-2)\pi}{N_s}} + R_1 e^{\frac{-j2(N_s-2)\pi}{N_s}}) \quad \text{Equation 22}$$

[0047] $\Lambda^{(N_s)}$ is per Equation 23.

$$\Lambda^{(N_s)} = (R_0 + R_1^H e^{\frac{j2(N_s-1)\pi}{N_s}} + R_2^H e^{\frac{j4(N_s-1)\pi}{N_s}} + R_2 e^{\frac{-j4(N_s-1)\pi}{N_s}} + R_1 e^{\frac{-j2(N_s-1)\pi}{N_s}}) \quad \text{Equation 23}$$

Although Equations 17-23 illustrate using the first row of $R_C D$ and $D \Lambda$, any row can be used to determine the $\Lambda^{(i)}_s$.

[0048] To illustrate using a center row, $(N_s/2)^{\text{th}}$ row (row 5 of Equation 7), $\Lambda^{(1)}$ is per Equation 19.

$$\Lambda^{(1)} = (R_0 + R_1^H + R_2^H + R_1 + R_2) \quad \text{Equation 19}$$

Equations 19-23 compute the FFT of K by K blocks. Since these blocks are multiplied by scalar exponentials, this process is referred to as a "block FFT." Typical approaches for calculating FFTs, such as the Matlab function "fft" compute the FFTs of a one-sided sequence. Since each $\Lambda^{(i)}$ is a two sided sequence, the computation of $\Lambda^{(i)}$ can be implemented by a fft $\{0,0,\cdots,R_2,R_1,R_0,R_1^H,R_2^H,\cdots,0,0\}$ and multiplying it by an appropriate exponential function as per Equation 27 for a center row.

$$e^{j2\pi(k-1)\nu}, \text{ where } \nu = \frac{\left[ceil\left(\frac{N_s}{2}\right)-1\right]}{N_s} \quad \text{Equation 27}$$

[0049] As shown in Equations 17-27, calculating all the $\Lambda^{(i)}$ can be performed using a single column of R. Accordingly, R_c is not required to be determined. Any column of R can be used to derive the $\Lambda^{(i)}$ s directly. Preferably, a row at least W rows from either edge of R is used, since these rows have a full set of R_i s.

[0050] Using the $\Lambda^{(i)}$ s and the D matrix, the block-circulant matrix R_c can be re-written as Equations 28 and 29.

$$R_c D = D \Lambda \quad \text{Equation 28}$$

$$R_c = \left(1/N_s\right) \left[D \Lambda D^H \right] \quad \text{Equation 29}$$

D and Λ are each of size (KN_s) by (KN_s) .

[0051] Since $D^H D = N_s I_{KN_s}$, $D^{-1} = (1/N_s)D^H$, Equation 30 results.

$$R_c^{-1} = N_s [(D^H)^{-1} \Lambda^{-1} (D)^{-1}] = N_s \left[\frac{D}{N_s} \Lambda^{-1} \frac{D^H}{N_s} \right] \quad \text{Equation 30}$$

[0052] The MMSE solution is per Equation 31.

$$\underline{d} = R_c^{-1} (A^H \underline{r}) \quad \text{Equation 31}$$

The detected data vector \underline{d} is of size $(N_s K)$ by 1.

[0053] The MMSE solution is per Equation 32.

$$D^H \underline{d} = \Lambda^{-1} \left[D^H (A^H \underline{r}) \right] \quad \text{Equation 32}$$

The matrix Λ is of size (KN_s) by (KN_s) with K by K blocks, and its inverse is per Equation 33.

$$\Lambda^{-1} = \begin{bmatrix} \Lambda^{(1)} & & & & \\ & \Lambda^{(2)} & & & \\ & & \Lambda^{(3)} & & \\ & & & \ddots & \\ & & & & \Lambda^{(N_s)} \end{bmatrix}^{-1} = \begin{bmatrix} [\Lambda^{(1)}]^{-1} & & & & \\ & [\Lambda^{(2)}]^{-1} & & & \\ & & [\Lambda^{(3)}]^{-1} & & \\ & & & \ddots & \\ & & & & [\Lambda^{(N_s)}]^{-1} \end{bmatrix}$$

Equation 33

The inversion requires an inversion of the K by K matrices $\Lambda^{(k)}$.

[0054] As a result, the data vector \underline{d} is determined per Equation 34.

$$[F(\underline{d})]_k = [\Lambda^{(k)}]^{-1} [F(A^H \underline{r})]_k \quad \text{Equation 34}$$

Equation 34 is applicable to both receivers which sample the received signal at the chip rate and at a multiple of the chip rate, such as twice the chip rate. For multiple chip rate receivers, the R matrix corresponding to the multiple chip rate is of the same form as Equation 9, being approximately block circulant.

[0055] To reduce the complexity in determining $F(A^H \underline{r})$, an FFT approach taking advantage of the structure of A may be used. A has an approximately block-circulant structure. However, it is a non-square matrix, being of size $(N_s \text{SF})$ by $(N_s \text{K})$. An illustration of a matrix A is per Equation 35.

$$A = \begin{bmatrix} \begin{bmatrix} b_1^{(1)}(0) & b_1^{(K)}(0) \\ b_{SF}^{(1)}(0) & b_{SF}^{(K)}(0) \end{bmatrix} & O & O & \dots \\ \begin{bmatrix} b_1^{(1)}(1) & b_1^{(K)}(1) \\ b_{SF}^{(1)}(1) & b_{SF}^{(K)}(1) \end{bmatrix} & \begin{bmatrix} b_1^{(1)}(0) & b_1^{(K)}(0) \\ b_{SF}^{(1)}(0) & b_{SF}^{(K)}(0) \end{bmatrix} & O & \dots \\ \begin{bmatrix} b_1^{(1)}(2) & b_1^{(K)}(2) \\ b_{SF}^{(1)}(2) & b_{SF}^{(K)}(2) \end{bmatrix} & \begin{bmatrix} b_1^{(1)}(1) & b_1^{(K)}(1) \\ b_{SF}^{(1)}(1) & b_{SF}^{(K)}(1) \end{bmatrix} & \begin{bmatrix} b_1^{(1)}(0) & b_1^{(K)}(0) \\ b_{SF}^{(1)}(0) & b_{SF}^{(K)}(0) \end{bmatrix} & \dots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

Equation 35

Each $b_j^{(k)}(i)$ is the convolution of the channel response $h^{(k)}$ and the spreading code $c^{(k)}$, for the k^{th} user at the j^{th} chip interval of the i^{th} symbol interval.

[0056] Using blocks $B(\cdot)$, where each block is indicated by bracketing in Equation 35, Equation 35 becomes Equation 36.

$$A = \begin{bmatrix} B(0) & 0 & 0 & 0 & \dots & 0 \\ B(1) & B(0) & 0 & 0 & \dots & 0 \\ B(2) & B(1) & B(0) & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & B(2) & B(1) & B(0) \\ \vdots & \vdots & & \vdots & \vdots & \vdots \end{bmatrix} \quad \text{Equation 36}$$

[0057] As shown, a portion of A is block-circulant. A circulant extension of A is denoted as A_c .

A can be broken into three matrices per Equation 37.

$$A = D_1 \Lambda_1 D_2^H \quad \text{Equation 37}$$

D_1 is a $(N_s SF)$ by $(N_s SF)$ matrix. D_2 is a $(N_s K)$ by $(N_s K)$ matrix and Λ_1 is a block-diagonal matrix of size $(N_s SF)$ by $(N_s K)$.

[0058] The block diagonal matrix Λ_1 has the same form as Λ of Equation 14. However, each of its entries $\Lambda_1^{(i)}$ is a SF by K block per Equation 38.

$$\Lambda_1^{(i)} = \begin{bmatrix} \lambda_{1,1}^{(i)} & \dots & \lambda_{1,K}^{(i)} \\ \vdots & & \vdots \\ \lambda_{SF,1}^{(i)} & \dots & \lambda_{SF,K}^{(i)} \end{bmatrix}.$$

$$\quad \text{Equation 38}$$

[0059] D_2 is the same form as that of D in Equation 11. D_1 is of the form of Equation 39.

$$D_1 = \begin{bmatrix} I & e^{\frac{j2\pi}{N_s}} I_{SF} & e^{\frac{j4\pi}{N_s}} I_{SF} & \dots & \dots & e^{\frac{j18\pi}{N_s}} I_{SF} \\ I & e^{\frac{j4\pi}{N_s}} I_{SF} & e^{\frac{j8\pi}{N_s}} I_{SF} & \dots & \dots & e^{\frac{j36\pi}{N_s}} I_{SF} \\ \vdots & \vdots & \vdots & & & \vdots \\ I & e^{\frac{j18\pi}{N_s}} I_{SF} & e^{\frac{j36\pi}{N_s}} I_{SF} & \dots & \dots & e^{\frac{j162\pi}{N_s}} I_{SF} \\ I & e^{\frac{j20\pi}{N_s}} I_{SF} & e^{\frac{j40\pi}{N_s}} I_{SF} & \dots & \dots & e^{\frac{j180\pi}{N_s}} I_{SF} \\ I & e^{\frac{jN_s\pi}{N_s}} I_{SF} & e^{\frac{jN_s\pi}{N_s}} I_{SF} & \dots & \dots & e^{\frac{jN_s\pi}{N_s}} I_{SF} \end{bmatrix}$$

Equation 39

I_{SF} is a SF by SF identity matrix.

[0060] In multiplying A_C and D_2 , products of the form, $B(i)$ and $e^{\frac{j2\pi}{N_s}} I_K$, are formed per

Equation 40.

$$A_C D_2 = \begin{bmatrix} [B(0) + B(1) + B(2)] & [B(0)e^{\frac{j2\pi}{N_s}} + B(2)e^{\frac{j18\pi}{N_s}} + B(1)e^{\frac{j20\pi}{N_s}}] & \dots & [B(0)e^{\frac{j18\pi}{N_s}} + B(2)e^{\frac{j162\pi}{N_s}} + B(1)e^{\frac{j180\pi}{N_s}}] \\ [B(0) + B(1) + B(2)] & [B(1)e^{\frac{j2\pi}{N_s}} + B(0)e^{\frac{j4\pi}{N_s}} + B(2)e^{\frac{j20\pi}{N_s}}] & \dots & [B(1)e^{\frac{j18\pi}{N_s}} + B(0)e^{\frac{j36\pi}{N_s}} + B(2)e^{\frac{j180\pi}{N_s}}] \\ \vdots & \vdots & & \vdots \\ [B(0) + B(1) + B(2)] & [B(2)e^{\frac{j16\pi}{N_s}} + B(1)e^{\frac{j18\pi}{N_s}} + B(0)e^{\frac{j20\pi}{N_s}}] & \dots & [B(2)e^{\frac{j144\pi}{N_s}} + B(1)e^{\frac{j162\pi}{N_s}} + B(0)e^{\frac{j180\pi}{N_s}}] \\ \vdots & \vdots & & \vdots \end{bmatrix}$$

Equation 40

$A_c D_2$ is of size $(N_s SF)$ by $(N_s K)$ and each block is a size SF by K.

[0061] In multiplying D_1 and Λ_1 , products of the form, $e^{\frac{j2\pi}{N_s}} I_{SF}$ and $\Lambda_1^{(i)}$, are formed.

$D_1 \Lambda_1$ is of size $(N_s SF)$ by $(N_s K)$ and each block is of size SF by K. Comparing any row of

$A_c D_2$ with the same row of $D_1 \Lambda_1$, Equation 41 results.

$$\begin{aligned}\Lambda_1^{(1)} &= [B(0) + B(1) + B(2)], \\ \Lambda_1^{(2)} &= [B(0) + B(1)e^{\frac{-j2\pi}{N_s}} + B(2)e^{\frac{-j4\pi}{N_s}}], \\ &\vdots \\ \Lambda_1^{(N_s-1)} &= [B(0) + B(1)e^{\frac{-j2(N_s-2)\pi}{N_s}} + B(2)e^{\frac{-j4(N_s-2)\pi}{N_s}}], \\ \Lambda_1^{(N_s)} &= [B(0) + B(1)e^{\frac{-j2(N_s-1)\pi}{N_s}} + B(2)e^{\frac{-j4(N_s-1)\pi}{N_s}}].\end{aligned}$$

Equation 41

[0062] As a result, each $\Lambda_1^{(k)}$ can be determined using the FFT of a one-sided sequence of $(SF \text{ by } K)$ blocks. Using Equation 38 and $D_2^H D_2 = N_s I_{KN_s}$, Equations 42, 43 and 44 result.

$$A = D_1 \Lambda_1 D_2^H, \quad \text{Equation 42}$$

$$A^H \underline{r} = D_2 \Lambda_1^H (D_1^H \underline{r}) \quad \text{Equation 43}$$

$$D_2^H \left(A^H \underline{r} \right) = N_s \cdot \left[\Lambda_1^H \left(D_1^H \underline{r} \right) \right] \quad \text{Equation 44}$$

[0063] Accordingly, $\left[F \left(A^H \underline{r} \right) \right]_k$ is determined using FFTs per Equation 45.

$$\left[F \left(A^H \underline{r} \right) \right]_k = N_s \cdot \left[\Lambda_1^{(k)} \right]^H \left[F \left(\underline{r} \right) \right]_k \quad \text{Equation 45}$$

Similarly, since the matrix A is approximately block-circulant, $R = A^H A + \sigma^2 I$ can also be computed using FFTs using Λ_1 .

[0064] To reduce complexity, the inversion of each $\Lambda^{(i)}$, $\left[\Lambda^{(i)} \right]^{-1}$, can be performed using LU decomposition. Each $\left[\Lambda^i \right]$ is a (K by K) matrix whose LU decomposition is per Equation 46.

$$\Lambda^{(i)} = LU \quad \text{Equation 46}$$

[0065] L is a lower triangular matrix and U is an upper triangular matrix. Equation 7 is solved using forward and backward substitution per Equations 47 and 48.

$$\left[\Lambda_g^{(k)} \right] y = \left[F \left(A^H \underline{r} \right) \right]_k \quad \text{Equation 47}$$

$$y = \left[\Lambda_g^{(k)} \right]^H \left[F \left(\underline{d} \right) \right]_k \quad \text{Equation 48}$$

[0066] Preferably, to improve the bit error rate (BER) for data symbols at the ends of each data fields 22, 24, samples from the midamble portion 20 and guard period 18 are used in the data detection as shown in Figure 5. To collect all the samples of the last symbols in the data fields, the samples used to determine r are extended by $W-1$ chips (the length of the impulse response) into the midamble 20 or guard period 18. This extension allows for substantially all the multipath components of the last data symbols of the field to be used for data detection. As shown for data field 1 22, the samples are extended into the midamble by $W-1$ chips. The midamble sequences are cancelled from the samples taken from the midamble 20 prior to data detection processing. For data field 2 24, the samples are extended into the guard period 18 by $W-1$ chips.

[0067] Certain FFT implementations required a certain field length for analysis. One of these FFT implementations is a prime factor algorithm (PFA). The PFA implementation requires the field length to be a prime number, such as 61. To facilitate PFA FFT implementation, the samples used to determine r are preferably extended to a desired PFA length. As shown in Figure 5, data field 1 or data field 2 are extended by P chips to the desired PFA length. Alternately, block FFTs of 61 symbols are extended to block FFTs of length 64, which requires 2^n FFT calculations. Since the approximation of R to a block circulant matrix is reduced, the performance typically improves.

[0068] An analysis of the computational complexity of fast joint detection is as follows. The computational complexity of calculating A is $K \cdot SF \cdot W$. The computational complexity of calculating $A^H A$ is per Equation 49.

$$\frac{(K^2 + K)}{2} [2(SF + W - 1) - (n_{\max} - 1)] \frac{n_{\max}}{2} - \frac{(K^2 - K)}{2} (SF + W - 1)$$

where $n_{\max} = \min(N_s, (SF + W - 1)/SF) + 1$ Equation 49

[0069] Calculating $(A^H \underline{r}) A$ as a matrix-vector multiplication has a computational complexity of $K N_s (SF + W-1)$. Calculating the FFT of a j^{th} column-block of R requires $K^2 \cdot (N_s \log N_s)$ calculations. Calculating the Fourier transform of $A^H \underline{r}$ requires $K(N_s \log_2 N_s)$ calculations. The inversion of each matrix $[\Lambda^{(k)}]$, without Cholesky decomposition, requires K^3 calculations. For N_s frequency points, the total number of calculations is $N_s K^3$. Calculating $[F(\underline{d})]_k = [\Lambda^{(k)}]^{-1} [F(A^H \underline{r})]_k$ requires (K^2) multiplications for N_s frequency points. Accordingly, the total number of calculations is $N_s K^2$. The Inverse FFT of $[F(\underline{d})]$ requires $K(N_s \log_2 N_s)$ calculations.

[0070] To illustrate the complexity for fast joint detection, the million real operations per second (MROPs) for processing a TDD Burst Type I with $N_c = 976$, $SF = 16$, $K = 8$, $N_s = 61$ and $W = 57$ chips is determined. The calculations A , $(A^H A)$, a column-block of R , $[\Lambda^{(k)}]^{-1}$ are performed once per burst, i.e. 100 times per second. The calculations $A^H \underline{r}$, $F[A^H \underline{r}]$, computing $[F(\underline{d})]_k$ and the inverse FFT of $[F(\underline{d})]$ are performed twice per burst, i.e. 200 times per second. Four calculations are required to convert a complex operation into a real operation. The results are illustrated in Table 1.

<u>Functions executed once per burst</u>	<u>MROPS</u>
Calculating A	3.0
Calculating $A^H A$	4.4
Calculating $F([R]_j)$	9.2614
Calculating $[\Lambda^{(k)}]^{-1}$	12.4928

Functions executed twice per burst

Calculating $A^H \mathbf{r}$	28.11
Calculating $F[A^H \mathbf{r}]$	2.3154
Calculating $[F(\hat{\mathbf{d}})]_k = [\Lambda^{(k)}]^{-1} [F(A^H \mathbf{r})]_k$	3.1232
Inverse FFT of $[F(\hat{\mathbf{d}})]$	2.3154
Total number of MROPS required for fast joint detection	65.0182

Table 1

Note: in Table 1, $(A^H \mathbf{r})$ was calculated directly as a matrix-vector multiplication.

[0071] If LU decomposition is used to determine $[\Lambda^{(k)}]^{-1}$, the complexity reduces to

54.8678 MROPS. If FFTs are used to determine $(A^H \mathbf{r})$, the complexity reduces from 65.0182 MROPS to 63.9928 MROPS.

[0072] A comparison of the complexity of fast joint detection and other detection techniques is as follows. The complexity of the following three techniques for a TDD Burst Type I with SF = 16 and K = 8 is per Table 2.

<u>Technique</u>	<u>MROPS</u>
Approximate Cholesky based Joint Detection, (JDChol)	82.7
Single User Detection: Approximate Cholesky based Equalization followed by a Hadamard Transform based Despreading (SDChol)	205.2276

Fast Joint Detection (JDFFT)	65.0182
------------------------------	---------

Table 2

[0073] The performance of the three detection techniques and a reference matched filtering (MF) data detection technique were compared by simulation over 800 timeslots. The simulations used the precision provided by Matlab, i.e. no finite precision effects were considered. The simulations used channels specified by WCDMA TDD WG4; SF = 16 and K = 8 and 12 and were performed for the downlink with no transmit diversity to facilitate comparison to SUD.

[0074] As shown in Figures 6 and 7, respectively, for cases 1 and 3, the performance of fast joint detection, JDFFT, is very close to that of the Cholesky based Joint Detection, JDChol. The other data detection approaches did not perform as well as JDChol or JDFFT. For the tddWg4Case2 channel as shown in Figure 8. JDFFT shows some degradation compared to JDChol. It also performs similarly to the SUD based Cholesky algorithm, SDChol. For a high data rate service, such as a 2 Mbps service, as shown in Figures 9-11. JDFFT performs close to or slightly worse than JDChol and better than the other approaches.

* * *